ECHO Client Partner User Guide - Chapter 3

ECHO has been replaced by the **Common Metadata Repository (CMR)**, a high-performance, high-quality, continuously evolving metadata system that catalogs all data and service metadata records for the EOSDIS system and will be the authoritative management system for all EOSDIS metadata.

The information contained within this ECHO wiki is now archived for historical reference. Please navigate to the **CMR wiki pages**, or to the **CMR Overview page** on **Earthdata**.

Logging in, setting up and getting started

This section contains information and examples that are common to most ECHO client applications, such as basic Login and Logout and creation and management of user accounts.

ECHO Tokens & User Contexts

There are two types of user in ECHO. Registered users and guests. Registered users can have access to restricted data and services but requires the creation of an Earthdata Login. Guest users do not require an Earthdata Login but their access is limited in some areas.

Echo tokens are generated whenever a user interacts with an ECHO API. A token is a transient representation of either your URS credentials or your guest status.

In both ECHO-REST and ECHO-SOAP APIs registered access requires the creation of an ECHO token and then use of that token throughout your interaction with ECHO.

In the case of ECHO-REST guest token creation is carried out behind the scenes of the API. In the case of ECHO-SOAP a client must manually create their token to interact with the API.

User Accounts

Create your user credentials in URS and use them to interact with ECHO using your user name and password.

Creating and Managing ECHO Sessions

When using the Web Services API, you need to request an **ECHO token** for all but ECHO-REST guest access.

This token acts as your session key and must be passed to all other ECHO operations. All clients interfacing with the system are required to pass client identifier information to ECHO. The client identifier is a short description and/or name of the client. Client developers are encouraged to keep this information as concise as possible and to work with the ECHO Operations Group to create an appropriate identifier.
A token is obtained when logging in to ECHO (see Logging In to ECHO). This is also done when setting user and provider context.
When done working in ECHO, be sure to logoff (see Logging Out of ECHO.).
NOTE: If client identifier information is not provided, submitted orders will fail.

Logging In to ECHO

The REST way

Create a token via a POST to the token resource

Creating an ECHO token for a user

```
Request headers:
Content-Type: application/xml

Request:
POST https://api.echo.nasa.gov/echo-rest/tokens

Request body:
<token>
 <username>Your URS username</username>
    <password>Your URS password</password>
    <client_id>An arbitrary ID to identify yourself</client_id>
    <user_ip_address>Your IP address</user_ip_address>
</token>
```

If you want to perform operations as a provider then you need the provider id and the credentials of a user that has provider privileges

Creating an ECHO token for a provider

```
Request headers:
Content-Type: application/xml

Request:
POST https://api.echo.nasa.gov/echo-rest/tokens
Request body:
<token>
 <username>Your URS username</username>
    <password>Your URS password</password>
    <client_id>An arbitrary ID to identify yourself</client_id>
    <user_ip_address>Your IP address</user_ip_address>
 <behalfOfProvider>Your provider ID</behalfOfProvider>
 </token>
```

The ECHO token will be returned to you as follows in the response body,

ECHO Token response

```
<?xml version="1.0" encoding="UTF-8"?>
<token>
  <id>ECHO-TOKEN-ID</id>
  <username>Your URS username</username>
  <client_id>An arbitrary ID to identify yourself</client_id>
  <user_ip_address>Your IP address</user_ip_address>
</token>
```

You will use the value of ECHO-TOKEN-ID to interact with the ECHO REST API.

The SOAP way

To obtain a token, call the Login operation of the Authentication Service. A new token is created and returned each time Login is invoked. Code Listing 1 is an example of logging in to ECHO and creating a session token for a guest user.
Parameters:

- username - Username of the user logging in
- password - Password of the user
- clientInfo - The string identifier of the ECHO client used to make this request
- actAsUserName - name of the user an Admin wants to act as, null for non ECHO administrator users
- behalfOfProvider - provider the user wants to act as , null for guests and registered users with no ProviderRoles

A Security Token is returned that is used for all subsequent calls to ECHO using the same Token profile.

Code listing 1: Logging in as Guest

```
// Client information is optional information provided by the
// client to ECHO when a user logs in
ClientInformation clientInfo = new ClientInformation();
clientInfo.setClientId("A Client");
clientInfo.setUserIpAddress("192.168.1.1");
// Call login with guest as username, email as password, and
// client information
String token = authenticationService.login("guest", "john@doe.com",
clientInfo, null, null);
```

Code Listing 2 is an example of logging in to ECHO and creating a session token for a guest user.

Code Listing 2: Logging In as a Registered User

```
// Client information is optional information provided by the
// client to ECHO when a user logs in
ClientInformation clientInfo = new ClientInformation();
clientInfo.setClientId("A Client");
clientInfo.setUserIpAddress("192.168.1.1");
// Call login with jdoe as username, mypass as password, and
// client information
String token = authenticationService.login("jdoe", "mypass", clientInfo,
null, null);
```

Logging Out of ECHO

The REST way

To logout of your session simply delete your ECHO token.

Logging out - deleting an ECHO token

```
DELETE https://api.echo.nasa.gov/echo-rest/tokens/ECHO-TOKEN-ID
Response: Status Code: 204 No Content
```

The SOAP way

When you are finished with a token, for example, you have finished a session with ECHO; destroy the token using the Logout operation. Parameter(s):

- token - security token

Code Listing 3: Logging out of ECHO

```
// Logout and set token to null because it is not useful anymore
authenticationService.logout(token);
token = null;
```

You do not need to destroy a token after each call; you may reuse a token until it is destroyed with the Logout operation or the token expires. Because the token is used to track your session, it must be protected by client applications with the same level of security that you use for your login name and password

Interacting with ECHO

The REST way

You can interact with any ECHO-REST resource using guest by simply GETing, POSTing, PUTing or DELETEing the resource. Some of these operations will fail if guest does not have the authority to perform them. In cases, where you want to use a registered user you should acquire an ECHO token above and attach it as a header to the request you make. For example,

Getting a list of providers with an ECHO token

```
Request headers:
Content-Type: application/xml
Echo-Token: ECHO-TOKEN-ID

Request:
GET https://api.echo.nasa.gov/echo-rest/providers

Response headers:
Status Code: 200 OK

Response Body:
<?xml version="1.0" encoding="UTF-8"?>
<providers type="array">
  <provider>
   ...
  </provider>
</providers>
```

The SOAP way

Interacting with the rest of the ECHO API follows the same pattern as logging in and logging out except that it requires that you pass a valid ECHO token to each operation. The following example shows logging in, retrieving the version number of the ECHO system and logging back out.

Code Listing 4: Getting the ECHO Version Number

```
// Login
String token = authenticationService.login("jdoe", "mypass", new
ClientInformation("A Client", "192.168.1.1"), null, null);
// Print out echo version number.
System.out.println("ECHO's version number: " +
authenticationService.getECHOVersion(token));
// Logout using token from previous login
authenticationService.logout(token);
token = null;
```